

Generating Autonomous Dynamic Behavior for Computer Animation: A Constrained Optimal Control Approach

Sunil K. Singh Alok Kumar Lin Shi
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755-8000.

1 Dynamic Behavior and Sensorimotor Coordination

Recently there has been considerable emphasis on *autonomous virtual actors* capable of generating their own actions in response to a desired goal and prescribed behavior [14]. While such actors may be described by purely kinematic descriptions, realistic animations require their descriptions to include the dynamics as well as all dynamic constraints. A similar approach has been attempted for much simpler tasks using the subsumption architecture [1] which decomposes an artificial creature's abilities into several parallel task-achieving behaviors. This framework cannot deal with systems with the following characteristics:

- Coordination of systems whose input-output relationships are expressed by ordinary differential equations and where sub-systems have nonlinear dynamic coupling.
- Constraints on available torques/forces.
- A highly dynamic environment.

In this paper we outline a framework that would enable us to program complex dynamic sensorimotor behavior in such characters. Since the emphasis is on *dynamic* behaviors, it is necessary to deal with and reason about dynamic variables such as velocities and accelerations; hence a dynamical systems approach is adopted. Our approach is to demonstrate that apparently complex patterns of dynamic coordination (such as obstacle avoidance, pole-balancing, jumping and hopping) can be formulated using a *dynamic constraint-generation and constraint satisfaction framework*. The lowest level motor controller *behavior* is altered by constraints generated by the supervisory layer on top. The solution to the resulting constrained problem, which is accomplished in real-time, automatically generates a *desired behavior pattern*. A central theme of the proposed research is the dynamic generation of sub-goals and constraints to satisfy each individual behavior.

We denote by w_i the *action variable*, which is the variable to be regulated while satisfying a desired behavior. During pole balancing, this is θ_i , the pole inclination; while avoiding an obstacle, it is d_i , the distance from the obstacle. Dynamic models of the action variable are regulated at the highest level by the *perception layer*. A central issue is that of the most *appropriate* yet computationally tractable dynamic models for *reasoning* about the dynamic behavior of the system. While the physical systems may be nonlinear in behavior (as defined by the input-output properties), the *desired behavior* of the perceived (action) variables may be described by a simple linear differential equation, usually a second order oscillator model given by

$$\ddot{w}_i + 2\zeta\omega_n\dot{w}_i + \omega_n^2 w_i = \omega_n^2 r \quad (1)$$

where w_i is the action variable and ζ and ω_n are parameters that characterize the dynamic behavior of the variable. There is some evidence [2] to support this description.

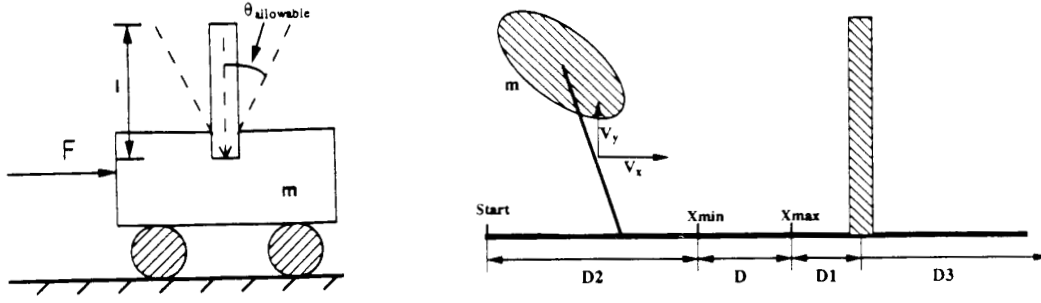


Figure 1: (a) Cart-pendulum system (b) Planar one-leg hopping robot.

1.1 A description of some dynamic behaviors

We have dealt with some typical dynamic behaviors. These include:

- (1) *Goal Reaching with Obstacle Avoidance*: For goal reaching we use a planar cart ($\bar{x} \in R^2$) moving towards a goal \bar{x}_f . The dynamic model in each direction is described by

$$\ddot{x}_i + 2\zeta_i\omega_i\dot{x}_i + \omega_i^2x_i = \omega_i^2r_i \quad (2)$$

and the objective is to find r_i such that the cart reaches the goal while avoiding the polyhedral obstacles.

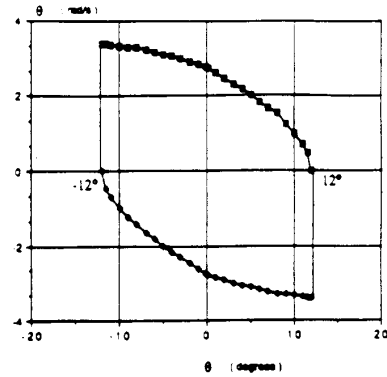
- (2) *Pole Balancing*: An interesting illustration of our approach is the study of the well-known *pole on a cart* problem shown in Figure 1(a). The only control that can be exerted is a linear force on the cart. The objective is to move the cart from the initial position x_0 to the desired final position x_f while keeping the pole balanced.
- (3) *Hopping*: Figure 1(b) shows the model we used for analyzing and simulating the hopping behavior. The key control variables are the horizontal and vertical velocities (V_x, V_y) of the robot during take-off.

2 The Proposed Framework for Sensorimotor Coordination and Dynamic Behavior

2.1 The Basic Solution Strategy

The overall solution procedure proposed in this work is summarized by the prediction-modification approach shown in Figure 2(a). The main steps involved are:

1. The *perception layer* generates the *unconstrained* desired trajectory. Our overall approach is summarized by the prediction-modification approach shown in Figure 2(a).
2. Predict, by looking into the future, whether the desired motion is safe. This is accomplished using *forward projections* (section 3.1).
3. Generate a set of permissible states (P-Set) using *preimage* computation (section 3.2).
4. If the desired state is not permissible the modifier uses the optimization routine to compute an optimal state which is closest to the desired state and is inside the permissible region (section 3.4).



5. A new constraint set is generated at each time instant, based on the current state of the system and the perception of the environment.

3.1 Prediction using Forward Projection

$$\dot{\bar{x}} = f(\bar{x}, \bar{u}) \quad \bar{x} \in \bar{R}^n$$

For the given system, forward projection is used to generate the *necessary conditions* for preventing failure. It predicts the state of the system at $t = \Delta t$ and in the future, and checks to see if the desired control is permissible. The current state at $t = 0$ is said to be permissible if by using the *command input* control $\bar{u} = \bar{k}(x)$ at $t = 0$ the system reaches a state at $t = \Delta t$ which does not violate any constraint. Further, if the optimal control u_{opt} is applied during the interval $(\Delta t, t_f)$, then the system safely reaches the sub-goal state that characterizes the particular behavior.

Let us consider the case of one dimensional pole balancing. The sub-goal set can be constructed as

$$Y_q = \{\theta, \dot{\theta} : \|\theta\| \leq \theta_{max}, \dot{\theta} = 0\}$$

$$k_2 - \frac{2\|k_1\|(\theta_{max} - \theta(t) - \dot{\theta}(t)\Delta t)}{\Delta t^2} \geq u \geq k_2 + \frac{2\|k_1\|(\theta_{max} + \theta(t) + \dot{\theta}(t)\Delta t)}{\Delta t^2}$$

The control u must satisfy the condition stated above and this would guarantee that $x_f \in Y_g$.

Obstacle Avoidance

For the planar cart described before, we examine the design of the *predictor*, which acts as the reflex controller checking if a commanded acceleration is *permissible*. Several autonomous navigation problems may be cast into a form where the obstacle constraints are characterized by the perception of a distance and the rate of change of this distance from a stationary or moving point (or plane). We examine a planar object ($\bar{x} \in R^2$) moving toward the plane $\bar{n}^T \bar{x} + \bar{c}$. If each Cartesian model is described by (2) then the time history of the distance between the point and the obstacle may be obtained as:

$$\ddot{d}(t) + 2\zeta\omega_n \dot{d}(t) + \omega_n^2 d(t) = \omega_n^2 \frac{\bar{n}^T \bar{r} + c}{\|\bar{n}\|} \quad (3)$$

assuming that the reference model in (2) has been chosen identical in each direction. Since the time history of the distance between the point and the plane is described by a second order system, we can analytically solve the unforced response for such a system to get the necessary conditions for predicting a collision free path. This condition requires $d > 0$ when $\dot{d} \leq 0$ [11].

3.2 Generation of constraints based on future outcomes: the computation of preimages

While prediction requires forward projection from a given state and helps us determine whether a *given state* is permissible, a much more useful description for the planner is the *set of all states (or initial conditions) from which a given goal or sub-goal set is reachable*. We shall refer to this set as the *preimage* of the desired state. Preimages generate the *sufficient conditions* for preventing failure. The definition of a preimage may be stated more formally as follows:

Given the desired goal state or a desired goal set Y_g and a desired constraint set Y_c , determine the set of all initial conditions, \bar{x}_c , which would guarantee that there exists some time $t_f \in (0, \infty)$ such that $\bar{x}(t_f) \in Y_g$ and $\bar{x}(t) \in Y_c \forall t \in (0, t_f]$.

From a control theory perspective, the preimages corresponds to the determination of the *domain of attraction* of the desired state or the desired state (equilibrium point). If $U(t)$ is *optimal*

$$\bar{x}(0) \in Y_{perm} \Rightarrow \forall t \quad \bar{x}(t) \in Y_{perm}$$

Application of this approach to the planar object avoiding a plane above would lead to the determination of the the preimage of the safe state by determining the critical condition that would permit the avoidance of collision. This set can be obtained as

$$\Pi = \left\{ \frac{\bar{n}^T \bar{r} + c}{\|\bar{n}\|} \geq \xi_{max}^* \right\} \quad (4)$$

Here ξ_{max}^* is the solution of following nonlinear algebraic equation

$$\xi_{max}^* + c_1(\xi_{max}^*) \exp[s_1 t_{min}^*(\xi_{max}^*)] + c_2(\xi_{max}^*) \exp[s_2 t_{min}^*(\xi_{max}^*)] = 0 \quad (5)$$

where ξ_{max}^* and t_{min}^* are obtained at this sampling instant $v(t = 0 \text{ or } d(0) \text{ and } \dot{d}(0))$. Therefore ξ_{max}^* is state dependent. Notice that the constraints are all linear in the control \bar{r} and convex. A real-time implementation of this has been discussed in [11].

The construction of the primages of the safe equilibrium state for the pole balancing problem is not a trivial task mainly because the system is so nonlinear. Figure 2(b) shows the preimage of the safe equilibrium state (0,0). This was generated using the nonlinear model described above [13].

3.3 The Decision Making Unit: The Predictor-Modifier Strategy

Algorithm 1 : The algorithm for prediction-modification scheme:

- Step 1. Based on the final goal state specification the perception layer constructs the desired behavior. Limit u_{des} using the hard control constraints.*
- Step 2. Identify the sub-goal state (or set) for the system and use forward projection to obtain the predicted state at $t = \Delta t$ and check if the desired control u_{des} is permissible.*
- Step 3. Check the sufficient conditions using the P-Set (preimages of the safe states). If $\mathbf{x} \in Y_{perm}$, the sufficient conditions will be satisfied, apply $F = F_{des}$, and go to Step 6.*
- Step 4. Solve the optimal control problem to obtain the modified optimal state of the system.*
- Step 5. Use the modified state to calculate new input to the system, $u = u_{modified}$.*
- Step 6. Go to the next time step (Step 1).*

Obstacle Avoidance

For certain behaviors, instead of using the states of the system directly it is more convenient to use a suitable function of the state in the decision process. The necessary and the sufficient conditions are also formulated using this function. In the case of obstacle avoidance the concepts of the *influence function* and the *modifier function* are used to implement the decision making strategy. The influence function is defined as:

$$D = d - \dot{d}\Delta t - \frac{\ddot{d}_{max}\Delta t^2}{2} \quad (6)$$

where \ddot{d}_{max} is the linear acceleration of the system when the maximum control force u_{max} is applied and d is the real distance from the obstacle.

Hopping

For the hopping robot the key control decisions for each hopping cycle are the horizontal and the vertical velocities (V_x, V_y) at take-off. The region surrounding the obstacle is divided into several domains (see Figure 1(b)) and the decision making unit uses these domains to reason whether or not to modify the state of the system. Let x_n denote the horizontal position of the robot during the n^{th} hopping cycle. The decision making unit reasons in the following manner:

- (1) If $x_n \in D_2$ and $x_{n+1} \in D_2$, do not consider the obstacles. Behave as if there were no obstacles.
- (2) If $x_n \in D_2$ and $x_{n+1} \in D$, the robot will avoid the obstacle. No need to modify the behavior.
- (3) If $x_n \in D_2$ and $x_{n+1} \in D_1$, modify V_x, V_y using the optimization routine.

3.4 Formulation of the Optimal Control Problem

In the previous sections we described how a particular behavior could be decomposed into the *goal*, the *desired mode of reaching the goal* and the *constraints that must be satisfied*. We then saw how these constraints could be interpreted as equality and inequality constraints within the proposed framework. The next step is to solve for the instantaneous course of action.

The most general solution to these problems is by solving a nonlinear mathematical programming problem or a Quadratic Programming problem if the constraints are linear. For a dynamical system with the following non linear dynamics

$$\dot{\bar{x}} = \bar{f}(\bar{x}, t) \quad \bar{x} \in R^n \quad (7)$$

the problem can be posed as a pointwise optimal control problem:

Minimize:

$$\|\ddot{\bar{x}}^d - \ddot{\bar{x}}\| \quad (8)$$

Subject to:

$$g(\bar{x}, \bar{u}) \leq 0 \quad (9)$$

In order to solve the optimization problem all the constraints must be translated to one single space, either the state space or the control space.

Obstacle Avoidance

For the planar object avoiding a plane, we obtain the following constraint satisfaction problem:

$$\min_{\bar{r} \in \Omega} \|\bar{r} - \bar{r}^d\| \quad (10)$$

where $\Omega = \{\Lambda \cap \Pi\}$, and $\Lambda = \{\bar{r} : \|\bar{r}\| \leq \bar{U}_{\max}\}$ and $\Pi = \{\frac{\bar{n}^T \bar{r} + c}{\|\bar{n}\|} \geq \xi_{\max}^*\}$. Notice that the constraints are all linear in the control \bar{r} and convex. Hence this is a linear QP problem which may be easily implemented in real-time [11].

Pole Balancing

For the one dimensional pole balancing behavior the optimization problem is solved in a slightly different manner using the *modification line* and the P-Set [13]. The modification line is an equivalent representation of the hard control constraints in the state space. The point of intersection of the modification line with the boundary of the P-Set defines an optimal state which satisfies all the constraints acting on the system.

Hopping

The optimization routine for the hopping behavior is formulated as follows:

Minimize:

$$(V_x - V_x^d)^2 + (V_y - V_y^d)^2 \quad (11)$$

Subject to:

$$g(V_x, V_y, H) = 0 \quad \|V_x\| \leq V_x^{\max} \quad \|V_y\| \leq V_y^{\max}$$

H is the desired hopping height of the robot. The constraint set is determined by the position of the obstacle, the available control force and the trajectory of the system during the flight phase.

4 Experimental Validation and Software Development

We have developed an extensive *simulator* to simulate the dynamics of the intelligent system. The simulation program was written in C and it runs on the IBM RISC/6000 workstations in an UNIX environment. A graphical user interface using NGI¹ was built and all the results were displayed graphically.

Figure 3(a) shows the result obtained in trying to avoid several planes. The implementation result for the pole balancing behavior is shown in Figure 3(b). Figure 3(c) shows the simulation result for the hopping motion when the maximum allowable acceleration during a hopping cycle is limited.

¹Northstar Graphics Interface

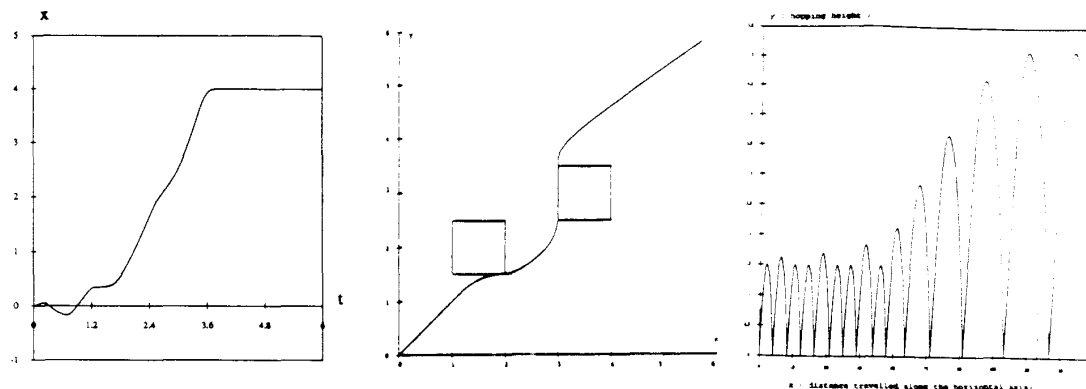


Figure 3: (a) The dynamic behavior of the planar object attempting to reach the goal while attempting to avoid the obstacles formed by the intersection of several planes. (b) Implementation of the pole balancer: linear position of the cart. (c) One dimensional hopping motion with limits on the control variable.

References

- [1] R. Brooks, "The Whole Iguana," *Robotics Science*, ed. M. Brady, MIT Press, 1989, pp. 432-456.
- [2] J. Kelso and A. Pandya, "Dynamic Pattern Generation and Recognition," *NSF Workshop on Mechanics, Control and Animation of Articulated Figures*, 1989.
- [3] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90-98 (1983).
- [4] B. H. Krogh, "Feedback Obstacle Avoidance Control," *Proceedings 21st Allerton Conference on Communication, Control and Computing*, Urbana, Illinois, October 1983, pp. 325-334.
- [5] R. Volpe and P. Khosla, "Manipulator Control With Superquadric Artificial Potential Functions: Theory and Experiments," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, Dec. 1990, pp. 1423-1436.
- [6] S. K. Kheradpir and J. S. Thorp, "Robust Real Time Control of Robot Manipulators in the Presence of Obstacles," *Proceedings of the IEEE Conference on Robotics and Automation*, Raleigh, NC, pp. 1146-1151 (1987).
- [7] W. S. Newman, "Automatic Obstacle Avoidance at High Speeds via Reflex Control," *Proceedings of the 1989 IEEE Conference on Robotics and Automation*, Scottsdale, Arizona, pp. 1104-1109 (1989).
- [8] S. K. Singh, "The ConSERTD Effort: A Control System Environment for Real-Time Decision-Making," *Proceedings of the IEEE Conference on Robotics and Automation*, Cincinnati, Ohio (1990).
- [9] T. Fukuda and Y. Kawauchi, "Cellular Robotic System (CEBOT) as one of the Realization of Self-Organizing Intelligent Universal Manipulator," *Proceedings of the IEEE Conference on Robotics and Automation*, Cincinnati, Ohio, 1990, pp. 662-669.
- [10] S.K. Singh, "Trajectory Planning for Robot Control: A Control Systems Perspective," *Advances in Control and Dynamic Systems*, Vol. 40, Academic Press, 1991, pp. 105-145.
- [11] L. Shi and S.K. Singh, "A Real-Time Collision Avoidance for a Point Object Amidst Polyhedral Obstacles," *Technical Report CAR-91-08, March, 1990*, Thayer School of Engineering, Dartmouth College, Hanover, NH.
- [12] S. K. Singh, "Trajectory Planning for Manipulators Under Uncertainty," *Proceedings of the Fourth IEEE International Symposium on Intelligent Control*, Albany, NY, pp. 152-158 (1989).
- [13] A. Kumar and S. K. Singh, "A Framework for Dynamic Task-Level Motion Planning," *Proceedings of the International Symposium on Intelligent Robotics*, Bangalore, India, Jan. 1993.
- [14] D. Zeltzer, "Motor Control Techniques for Figure Animation," *IEEE Computer Graphics and Applications*, November 1982, 2(9), pp. 53-59.